AMENDMENTS TO THE SPECIFICATION:

Please amend the specification as follows:

Please replace paragraphs [077], [093], and [094] with the following paragraphs:

[077] An illustrative circuit for use as adder unit 41 is described in U.S. Patent Applications Serial No. 10/035,595, filed on even date herewith in the name of Guy L. Steele Jr., and entitled "Floating Point Adder Circuit And Method", assigned to the assignee of the present application and hereby incorporated by reference. The exemplary adder unit 41 performs several types of operations, which may include but is not limited to addition of two operands, negation of one operand, subtraction of one operand from another operand, and absolute value of one operand. Generally, the negation operation is not affected by the rounding mode, and the result is always a copy of the operand with its sign reversed, even if the operand is in the NaN format 66. In the subtraction operation, the adder unit 41 generates the result as the sum of the operand that is the minuend and the negative of the operand that is the subtrahend. Essentially, the negative of the operand that is the subtrahend being the result of the negation operation. Effectively, the adder unit 41 performs a subtraction operation by performing a negation operation on the operand that is the subtrahend and an addition operation in connection with the operand that is the minuend and the result of the negation operation. The absolute value operation is also not affected by the rounding mode, and the result is a copy of the operand with its sign made positive, even if the operand is a NaN.

[093] An illustrative circuit for use as exemplary comparator unit 46 is described in U.S. Patent Application Serial No. 10/035,586, filed on even date herewith in the name of Guy L. Steel Jr., and entitled "Floating Point Comparator Circuit and Method," assigned to the assignee of the present application and hereby incorporated by reference. As noted above, exemplary comparator unit 46 receives two operands and generates a result that indicates whether the operands are equal, and, if not, which operand is larger. Generally, the comparison is not affected by the rounding mode. In a comparison operation consistent with an embodiment of the present invention,

two positive operands in the infinity format 65 are equal regardless of the values of the flags "nouzx,"

a positive operand in the infinity format 65 is greater than an operand in any other format, except for an operand that is in the NaN format 66,

two negative operands in the infinity format 65 are equal regardless of the values of the flags "nouzx,"

a negative operand in the infinity format 65 is less than an operand in any other format, except for an operand that is in the NaN format 66.

an operand in the NaN format 66 is unordered (*i.e.,* neither greater than, less than, nor equal to) another operand in any format, including another operand in the NaN format 66, and

operands in the format other than the infinity format 65 and NaN format 66 compare in accordance with IEEE Std. 754. Thus, +UN is greater than +0 and less than +TINY; +OV is greater than +HUGE and less than $+\infty$; and so on.

[094] An illustrative circuit for use as exemplary tester unit 47 is described in U.S. Patent Application Serial No. 10/035,741, filed on even date herewith in the name of Guy L. Steele Jr., and entitled "Floating Point Tester Circuit And Method," assigned to the assignee of the present application and hereby incorporated by reference. The tester unit 47 receives a single floating-point operand and determines whether it has one of a selected set of status conditions. Based upon the determination, the tester unit 47 produces a signal to indicate whether or not the selected condition holds. In one embodiment of the present invention, the conditions include:

the operand is in the NaN format 66;

the operand is in the infinity format 65;

the operand is in either the infinity format 65 or the NaN format 66;

the operand is in the overflow format 64;

the operand is in the overflow format 64 or contains a set overflow flag "o";

the operand is in the underflow format 61;

the operand is in the underflow format 61 or contains a set underflow flag "u";

the operand is in the zero format 60;

the operand is in the zero format 60 and the sign bit "s" is "zero" (representing value +0);

the operand is in the zero format 60 and the sign bit "s" is "one" (representing value –0);

the operand represents a non-zero number whose magnitude is less than 2^{-126};

the operand contains a set invalid-operation flag "n";

the operand contains a set divide-by-zero flag "z";

the operand represents a numerical value strictly between –OV and +OV;

the operand represents a numerical value strictly between -OV and +OV but is not +UN or -UN;

the operand is in any format 60 through 66 and contains a sign bit "s" that is "one"; and

all 11 non-trivial disjunctions ("OR") of subsets of:

a = (the operand is in overflow format 64 or contains a set overflow flag "o")

b = (the operand is in underflow format 61 or contains a set underflow flag "u")

c = (the operand contains a set invalid operation flag "n")

d = (the operand contains a set divide-by-zero flag "z")

In other words, the subsets include (a OR b), (a OR c), (a OR d), (b OR c), (b OR d), (c OR d), (a OR b OR c), (a OR b OR d), (a OR c OR d), (b OR c OR d) and (a OR b OR c OR d).